Title

A general framework for solving inverse dynamics problems in multi-axis motion control.

Journal

ISA transactions, 95

ISSN

0019-0578

Authors

Zhu, Bohan

Farouki, Rida T

Publication Date

2019-12-01

DOI

10.1016/j.isatra.2019.05.012

Peer reviewed

A general framework for solving inverse dynamics problems in multi-axis motion control

Bohan Zhu and Rida T. Farouki
Department of Mechanical and Aerospace Engineering,
University of California, Davis, CA 95616, USA.

Abstract

An inverse dynamics compensation (IDC) scheme for the execution of curvilinear paths by multi–axis motion controllers is proposed. For a path specified by a parametric curve $\mathbf{r}(\xi)$, the IDC scheme computes a real–time path correction $\Delta\mathbf{r}(\xi)$ that (theoretically) eliminates path deviations incurred by the inertia and damping of the machine axes. To exploit the linear time–invariant nature of the dynamic equations, the correction term is computed as a function of elapsed time t, and the corresponding curve parameter values ξ are only determined as the final step of the IDC scheme, through a real–time interpolator algorithm. It is shown that, in general, the correction term for P, PI, and PID controllers consists of derivative, natural, and integral terms (the integrand of the latter involving only the path $\mathbf{r}(\xi)$, and not its derivatives). The use of lead segments to minimize transient effects associated with the initial conditions is also discussed, and the performance of the method is illustrated by simulation results. The IDC scheme is expressed in terms of a linear differential operator formalism to provide a clear, general, and systematic development, amenable to further adaptations and extensions.

Keywords: CNC machine, PID controller, inverse dynamics, feedrate, contour error, path modification, Pythagorean–hodograph curve.

e–mail addresses: bzhu@ucdavis.edu, farouki@ucdavis.edu

1 Introduction

Precision motion control is a fundamental requirement in diverse application contexts, such as robotics, CNC machining, 3D printing, automated inspection, etc. In order to ensure the very tight geometrical tolerances required in the fabrication of engineering parts for superior functional efficiency, reliability, and durability, a very high motion accuracy of manufacturing equipment is essential. Feedback control plays a central role in this context, but optimization of the available control parameters may not suffice to achieve the desired accuracy in critical applications. The *inverse dynamics compensation* (IDC) method, proposed herein, aims to improve the performance of existing motion controllers by invoking a deterministic model of the internal machine dynamics to modify the nominal control input.

For high–speed execution of curvilinear paths by a computer numerical control (CNC) milling machine, the smoothing influence of inertia and damping of the machine axes can significantly compromise the accuracy of the actual path traversal. The IDC scheme builds upon existing feedback controllers by exploiting a dynamic model of the machine (obtained empirically by means of system identification software) to impose real–time modifications of the controller input that — subject to the accuracy of the dynamic model — will eliminate deviations of the executed path from the commanded path incurred by the machine physical limitations (axis inertia and damping). The control objective for the IDC scheme is to enhance high–speed tracking accuracy of curvilinear paths, by compensating for the known machine dynamics.

The present study develops a generalized approach to inverse dynamics compensation (IDC) schemes for free–form paths specified as parametric curves. The focus is on motion control systems with independently–controlled orthogonal linear axes, and the main physical effects of concern are the inertia and damping of the axes. A typical application is CNC machining, although the basic methodology is broadly applicable. The IDC scheme does not attempt to compensate for "disturbance" effects (machining forces, thermal expansion, etc.), that are not incorporated in the dynamical model. However, by eliminating the influence of known machine limitations[1] (axis inertia and damping) it minimizes the burden on the controller, allowing it to focus on compensating for unknown disturbances.

Prior studies [6, 20] approached the development of an IDC scheme for a path specified by a parametric curve $\mathbf{r}(\xi)$ by solving a system of differential equations, with the curve parameter ξ as independent variable. For a polynomial curve and P controller, a closed–form solution for the path correction term $\Delta\mathbf{r}(\xi)$ is possible. However, for PI or PID controllers, a closed–form solution is not possible, since the differential equations have polynomials in the parameter ξ as coefficients, and approximate solutions are undesirable since it can be difficult to guarantee that the approximation errors will always be much smaller than the path deviations that the IDC scheme is attempting to correct. The present study adopts a different IDC philosophy, wherein the path correction is expressed in terms of elapsed time t. This yields much simpler differential equations (with constant coefficients) that admit closed–form solutions for $\Delta\mathbf{r}(t)$, except for a single irreducible integral involving $\mathbf{r}(\xi)$ but not its derivatives.

[1] In high–speed machining [14, 22, 26] the axis inertia and damping may become the dominant limitations on path traversal accuracy.

The main contribution of this paper is to extend the closed–form solution for the IDC path correction term, as developed for a P controller in [6], to the case of PI and PID controllers. This is accomplished by delaying the essential — and in general non–trivial — transformation between time t and the curve parameter ξ to the final phase of the IDC scheme. Thus, the propagation of errors incurred in earlier phases is eliminated, yielding better accuracy of the path compensation term for PI and PID controllers (which obviate the initial positional discontinuities incurred by the path compensation term for a P controller). The conceptual framework and mathematical analysis underlying the extension of the IDC scheme to PI and PID controllers is developed herein, and is illustrated by some basic simulation results. The formulation of lead–in and lead–out segments, allowing motion along a curved path with smooth transitions from rest and to rest, is also developed. A follow–up paper will present more detailed experimental performance results from an implementation on a 3–axis CNC milling machine governed by a customized open–architecture software controller.

The plan for the remainder of the paper is as follows. In Sections 2 and 3 we introduce the basic machine axis dynamic model and PID controller adopted in this paper, and the problem of compensating for the influence of axis inertia and damping by the real–time modification of motion commands. For a curvilinear path $\mathbf{r}(\xi)$ specified in terms of a general parameter ξ, Section 4 discusses the key *real–time interpolation* problem of resolving the relation between the elapsed time t and location ξ along the path. Section 5 then introduces a linear differential operator formalism, that can be employed to correlate solutions to inverse dynamics models expressed in terms of t and ξ. The solutions to these problems are formulated in Section 6 for P, PI, and PID controllers in the time domain, exploiting the linear time–invariant (i.e., constant coefficient) nature of the differential equations. Section 7 then discusses the use of "lead" segments to minimize the transient effects associated with initial/final conditions for the differential equations. Finally, some representative simulation results are described in Section 8, and in Section 9 we recapitulate the main results of the present study and identify some possible extensions and generalizations of the IDC scheme.

2 Linear axis dynamic model

The linear axes of CNC machines are typically driven by electric motors through gear trains and ball screw mechanisms. Each axis has an associated inertia and damping, corresponding to the moving mass of the axis mechanism and viscous dissipation incurred by its lubrication. High–resolution position encoders on each axis allow the actual axis position to be measured in each sampling interval (typically $\sim 10^{-3}$ second) of the controller, and compared with the commanded position. The axis *position error* (i.e., the discrepancy between the commanded and measured position), is the basic control input to the axis motor.

Let $\mathbf{r}_d(t) = (x_d(t), y_d(t))$ and $\mathbf{r}_o(t) = (x_o(t), y_o(t))$ denote the desired (commanded) path and the output (actual) path executed by a CNC machine,[2] expressed in terms of elapsed time t. Standard models [1, 5] for CNC machine dynamics, assuming a PID controller, determine

[2]For brevity, we focus mainly on 2–axis motion here — the ideas generalize readily to 3–axis motion. Note that dots and primes denote derivatives with respect to time t and a general curve parameter ξ, respectively.

$\mathbf{r}_o(t)$ from $\mathbf{r}_d(t)$ through differential equations of the form

$$\begin{aligned} a_x \dddot{x}_o + b_x \ddot{x}_o + c_x \dot{x}_o + x_o &= d_x \ddot{x}_d + e_x \dot{x}_d + x_d \,, \\ a_y \dddot{y}_o + b_y \ddot{y}_o + c_y \dot{y}_o + y_o &= d_y \ddot{y}_d + e_y \dot{y}_d + y_d \,, \end{aligned} \tag{1}$$

where dots indicate time derivatives, and the constants $a_x, a_y, \ldots$ depend upon the machine physical and controller parameters. Figure 1 shows the block diagram of a typical axis drive system described by equations (1). The indicated quantities and their units are as follows: u is the control variable in V; k_a and k_t are the current amplifier and motor torque gains in A/V and N$\cdot$m/A; J and B are the axis inertia and viscous damping in kg$\cdot$m^2 and kg$\cdot$m^2/s; T, ω, and θ are the motor torque, angular speed, and position in N$\cdot$m, rad/s, and rad; and r_g is the transmission ratio (the axis translation per unit rotation of the motor shaft) in m/rad. Finally, k_p, k_i, k_d are the controller proportional, integral, and derivative gains.

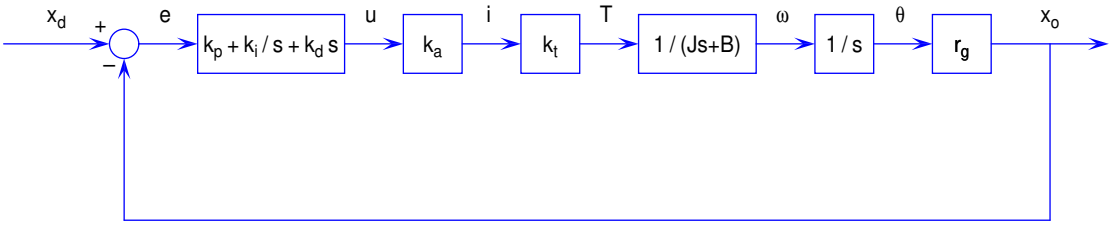

Figure 1: x–axis drive with PID controller gains k_p, k_i, k_d applied to position error $e = x_d - x_o$. The amplifier converts the control signal u into a current i to the motor, which exerts a torque T generating angular speed ω through the axis inertia J and damping B. Integrating ω yields the motor shaft angle θ, determining the axis position x_o through the transmission ratio r_g.

The coefficients for the x and y axis equations (1) may, in principle, be different because the physical quantities B, J, etc., may differ. We focus on the x–axis in the following sections, and omit axis subscripts (the same principles apply to the other axes). The approach may be generalized to distinct axes properties, and to the context of *cross–coupled control*, in which the axis motions are correlated [3, 4, 15, 16, 23, 25].

3 Inverse dynamics compensation

Setting $K = k_a k_t r_g$, the closed–loop transfer function relating the Laplace transforms of the x–axis output $X_o(s)$ and input $X_d(s)$ for the system shown in Figure 1 is

$$\frac{X_o(s)}{X_d(s)} = \frac{K(k_d\, s^2 + k_p\, s + k_i)}{(Js + B)s^2 + K(k_d\, s^2 + k_p\, s + k_i)}. \tag{2}$$

This defines a third–order system, with two zeros and three poles. The forward–path transfer function relating $X_o(s)$ and the position error $E(s) = X_d(s) - X_o(s)$, which is the same as the open–loop transfer function for a unity feedback control system, is

$$\frac{X_o(s)}{E(s)} = \frac{K(k_d\, s^2 + k_p\, s + k_i)}{(Js + B)s^2}. \tag{3}$$

To produce the exact desired position $X_d(s)$, we seek a modified commanded position $\hat{X}(s)$ or, equivalently, a compensatory displacement $\Delta X(s) = \hat{X}(s) - X_d(s)$ to be added to $X_d(s)$.

This "command modification" scheme amounts to solving an *inverse dynamics problem*, i.e., the determination of the required input for a given desired output (rather than determining an output from a given input). The transfer function relating $\Delta X(s)$ to $X_d(s)$ can be directly obtained from (3) by replacing $X_o(s)$, $E(s)$ by $X_d(s)$, $\Delta X(s)$ and taking the reciprocal, i.e.,

$$\frac{\Delta X(s)}{X_d(s)} = \frac{(Js + B)s^2}{K(k_d\, s^2 + k_p\, s + k_i)}. \tag{4}$$

The corresponding governing differential equations in the case of a P ($k_p \neq 0 = k_i = k_d$), PI ($k_p, k_i \neq 0 = k_d$), and PID ($k_p, k_i, k_d \neq 0$) controller are

$$\Delta x = \frac{1}{Kk_p}(J\ddot{x}_d + B\dot{x}_d), \tag{5}$$

$$\Delta\dot{x} + \frac{k_i}{k_p}\Delta x = \frac{1}{Kk_p}(J\dddot{x}_d + B\ddot{x}_d) \tag{6}$$

$$\Delta\ddot{x} + \frac{k_p}{k_d}\Delta\dot{x} + \frac{k_i}{k_d}\Delta x = \frac{1}{Kk_d}(J\dddot{x}_d + B\ddot{x}_d). \tag{7}$$

Note that the compact open–loop formulation (3) is only convenient for inverse problems, and equations (5)–(7) can be derived from (2) by simple algebra. In standard control problems, where $x_d(t)$ is a prescribed input, the transfer function (2) is usually incorporated in full. Moreover, the plant and controller dynamics are separable as the numerator and denominator in the open–loop transfer function (4) — compare with (2). They are clearly recognizable on opposite sides of the equations (5)–(7), suggestive of a physical interpretation.

4 General curve parameterizations

For a desired path defined by a differentiable curve $\mathbf{r}_d(\xi)$, the *parametric speed* $\sigma(\xi)$ specifies the rate of change of arc length[3] s with respect to the parameter ξ, namely

$$\sigma(\xi) = |\mathbf{r}'_d(\xi)| = \frac{\mathrm{d}s}{\mathrm{d}\xi}\,, \tag{8}$$

while the feedrate V is the derivative of arc length s with respect to time t,

$$V = \frac{\mathrm{d}s}{\mathrm{d}t}\,. \tag{9}$$

Since the dimensionless curve parameter ξ has no geometrical or physical meaning, feedrate is usually specified as either a constant, or as a function of a meaningful variable — such as

[3]Following usual practice, s denotes either the Laplace transform variable or the curve arc length. To avoid confusion, upper case characters indicate Laplace transforms of functions defined by lower case characters.

arc length s, path curvature κ, or time t [12]. For a prescribed path and feedrate, derivatives with respect to ξ and t are related by

$$\frac{\mathrm{d}}{\mathrm{d}t} = \frac{\mathrm{d}s}{\mathrm{d}t}\frac{\mathrm{d}\xi}{\mathrm{d}s}\frac{\mathrm{d}}{\mathrm{d}\xi} = \frac{V}{\sigma}\frac{\mathrm{d}}{\mathrm{d}\xi}. \tag{10}$$

In particular, the variation of the curve parameter ξ with time t is simply

$$\frac{\mathrm{d}\xi}{\mathrm{d}t} = \frac{\mathrm{d}s}{\mathrm{d}t}\frac{\mathrm{d}\xi}{\mathrm{d}s} = \frac{V}{\sigma}. \tag{11}$$

The ideal or "natural" parameterization [24] of a curve $\mathbf{r}_d(\xi)$ is that for which the parameter ξ coincides (up to a constant) with the arc length s, so that $\sigma \equiv 1$. However, no planar or spatial curve — other than a straight line — can be parameterized by simple (i.e., rational) functions of the arc length [9, 10], and a numerical solution of the relation (11) is therefore unavoidable. Some simplification is possible with the *Pythagorean–hodograph* (PH) *curves* [7] — a family of polynomial curves for which $\sigma(\xi)$ is a *polynomial* (rather than the square–root of a polynomial) in ξ, and hence the arc length $s(\xi)$ is also a polynomial in ξ.

In a feedback motion control system with sampling interval Δt, the controller compares the actual machine position $\mathbf{p}_k$ (measured by encoders on the machine axes) with the commanded machine position $\mathbf{r}_d(\xi_k)$, obtained by solving (11) for the curve parameter ξ_k at each sampling time $t_k = k\Delta t$, $k = 1, 2, \ldots$ The resulting position error $\mathbf{e}_k = \mathbf{r}_d(\xi_k) - \mathbf{p}_k$ is the basic input to the controller. The solution of (11) to generate a commanded machine position within each sampling interval is the function of the controller *real–time interpolator* module. Real–time interpolators for parametric curves are typically based on a truncated Taylor series expansion [2, 12, 13, 19, 21, 28] of the function $\xi(t)$ defined implicitly by (11) for a general polynomial curve, or closed–form reduction of certain integrals for a PH curve [8, 11, 27].

Since the parametric speed (8) is positive for a regular curve with $\mathbf{r}'_d(\xi) \neq \mathbf{0}$, and the feedrate V is also positive if no stopping or "back–tracking" is specified, the relation (11) indicates that ξ is a monotone–increasing function of t, and there is a unique ξ corresponding to each t. In general, however, a closed–form solution for the monotone function $\xi(t)$ is not possible, and numerical methods must be used to compute the reference–point parameter values ξ_k. A standard approach [12] is to compute ξ_{k+1} from ξ_k through a Taylor series of the form

$$\xi_{k+1} = \xi_k + \dot{\xi}(t_k)\Delta t + \frac{\ddot{\xi}(t_k)}{2!}(\Delta t)^2 + \frac{\dddot{\xi}(t_k)}{3!}(\Delta t)^3 + \cdots \tag{12}$$

Using the differentiation rule (10), the time derivative of ξ can be expressed [12] as

$$\dot{\xi} = \frac{V}{\sigma}, \quad \ddot{\xi} = \frac{\sigma V' - \sigma' V}{\sigma^2}\,\dot{\xi}, \quad \dddot{\xi} = \frac{\sigma V' - 3\sigma' V}{\sigma^2}\,\ddot{\xi} + \frac{\sigma V'' - \sigma'' V}{\sigma^2}\,\dot{\xi}^2, \quad \text{etc.},$$

where the parametric speed and its derivatives are

$$\sigma = |\mathbf{r}'_d|, \quad \sigma' = \frac{\mathbf{r}'_d \cdot \mathbf{r}''_d}{\sigma}, \quad \sigma'' = \frac{\mathbf{r}'_d \cdot \mathbf{r}'''_d + |\mathbf{r}''_d|^2 - \sigma'^2}{\sigma}, \quad \text{etc.}$$

Note that the feedrate V is, in general, either constant or specified as a function of a physical or geometrical variable (such as time t, arc length s, or curvature κ) and the derivatives V', V'', etc. with respect to ξ must be converted into derivatives with respect to these variables — the appropriate conversions may be found in [12].

The series expansion (12) — up to the second or third order term — is typically sufficiently accurate for a sampling frequency $f = 1$ kHz and sampling interval $\Delta t = 10^{-3}$ sec. For PH curves, further reduction of the relation (11) is possible, since $\sigma(\xi)$ is a positive polynomial, and the integration

$$s(\xi) = \int_0^{\xi} \sigma(\eta)\, \mathrm{d}\eta \tag{13}$$

yields a monotone–increasing polynomial as the arc length function. For a constant feedrate V_0, the relation (11) yields the equation

$$s(\xi_k) = kV_0\Delta t,$$

for the reference–point parameter values. This equation has a unique real root, which may be computed to machine precision by a few Newton–Raphson iterations with ξ_{k-1} as the starting value. Generalizing to a time–dependent feedrate $V(t)$ with integral $F(t)$, this becomes

$$s(\xi_k) = F(k\Delta t),$$

which is equally easy to solve. Finally, for a feedrate $V(s)$ dependent on the arc–length, one may reduce (11) to

$$\int_0^{s(\xi_k)} \frac{\mathrm{d}u}{V(u)} = k\Delta t,$$

where the integral on the left admits a closed–form reduction in certain simple cases. Further details may be found in [11, 27].

In the IDC scheme developed herein, the solution of the relation (11) for the parameter value ξ corresponding to any prescribed time t is executed on an as–needed basis, by calling the real–time interpolator function. This facilitates a more precise computation of the correction term in the context of PI and PID controllers.

5 Linear differential operator formalism

To discuss general properties of linear differential equations, and their solution structure, it is useful to introduce the notation

$$\begin{aligned} L_u\phi &= \left[\, p_n(u)D_u^n + p_{n-1}(u)D_u^{n-1} + \cdots + p_1(u)D_u + p_0(u) \,\right]\phi \\ &= p_n(u)\frac{\mathrm{d}^n\phi}{\mathrm{d}u^n} + p_{n-1}(u)\frac{\mathrm{d}^{n-1}\phi}{\mathrm{d}u^{n-1}} + \cdots + p_1(u)\frac{\mathrm{d}\phi}{\mathrm{d}u} + p_0(u)\phi, \end{aligned}$$

for an n^{th} order linear differential operator L_u acting on a function ϕ of the variable u, where the operator D_u denotes differentiation with respect to u. It is assumed here that $p_n(u) \neq 0$,

$p_{n-1}(u), \ldots, p_1(u)$, $p_0(u)$ are real–valued continuous functions on a sufficiently large open real interval I, and $\phi(u)$ has C^n continuity on I. Note that $L_u\phi$ is also a continuous function on I; its value at $u_0 \in I$ is denoted by $L_u\phi(u_0)$.

With this notation, the linear time–invariant (LTI) differential equations (5)–(7) assume the general form

$$L_t^c \Delta x(t) = L_t^d x_d(t), \tag{14}$$

L_t^c and L_t^d being polynomials in the time–derivative operator D_t, with constant coefficients, associated with the correction term $\Delta x(t)$ and the desired path $x_d(t)$. In the case (7) of the PID controller, for instance, we have

$$L_t^c = D_t^2 + \frac{k_p}{k_d} D_t + \frac{k_i}{k_d}, \quad L_t^d = \frac{1}{K k_d}(J D_t^3 + B D_t^2).$$

Note that, in each of the cases (5)–(7), the order of L_t^c is less than that of L_t^d.

By comparison, when an LTI system is expressed in terms of an independent variable other than time t, such as the curve parameter ξ or arc length s, the resulting differential operators acquire non–constant coefficients, that become increasingly complicated as the operator order increases. For example, through the change of variables $t \to \xi$, the third–order constant–coefficient differential operator

$$L_t = a_3 D_t^3 + a_2 D_t^2 + a_1 D_t + a_0 \tag{15}$$

in time t is transformed, through successive chain rule differentiation using (10), into the operator

$$L_\xi = \alpha_3 D_\xi^3 + \alpha_2 D_\xi^2 + \alpha_1 D_\xi + \alpha_0, \tag{16}$$

in the curve parameter ξ, where $\alpha_0 = a_0$ and the remaining *non–constant* coefficients are specified [12] by

$$\alpha_1 = \left[a_3 \left(\frac{\sigma V'' - \sigma'' V}{\sigma^2} \frac{V}{\sigma} + \frac{\sigma V' - 3\sigma' V}{\sigma^2} \frac{\sigma V' - \sigma' V}{\sigma^2} \right) + a_2 \frac{\sigma V' - \sigma' V}{\sigma^2} + a_1 \right] \frac{V}{\sigma},$$
$$\alpha_2 = \left(3 a_3 \frac{\sigma V' - \sigma' V}{\sigma^2} + a_2 \right) \left(\frac{V}{\sigma} \right)^2, \quad \alpha_3 = a_3 \left(\frac{V}{\sigma} \right)^3. \tag{17}$$

The change of variables from time t to arc length s is a special instance of the transformation $t \to \xi$, corresponding to $\sigma \equiv 1$. Then $L_t = a_3 D_t^3 + a_2 D_t^2 + a_1 D_t + a_0$ is transformed into the operator

$$L_s = \beta_3 D_s^3 + \beta_2 D_s^2 + \beta_1 D_s + \beta_0, \tag{18}$$

in arc length s, where $\beta_0 = a_0$ and the remaining coefficients

$$\beta_1 = \left[a_3 \left(V \frac{\mathrm{d}^2 V}{\mathrm{d}s^2} + \left(\frac{\mathrm{d}V}{\mathrm{d}s} \right)^2 \right) + a_2 \frac{\mathrm{d}V}{\mathrm{d}s} + a_1 \right] V, \quad \beta_2 = \left(3 a_3 \frac{\mathrm{d}V}{\mathrm{d}s} + a_2 \right) V^2, \quad \beta_3 = a_3 V^3$$

are non–constant (except in the case of a constant feedrate, $V \equiv V_0$ — when they are simply $\beta_i = a_i V_0^i$ for $i = 0, 1, 2, 3$).

The simplicity of the constant–coefficient differential operator (15) in time t, as compared to the complicated coefficients for the transformed operators (16) in the curve parameter ξ and (18) in arc length s, highlights the preferability of formulating and solving the inverse dynamics problem in the time domain, and finding the parameter value ξ_k corresponding to each sampling time $t_k = k\Delta t$ by solving the relation (11), for known variations of feedrate V and parametric speed σ, as a final step. Although equation (11) does not, in general, admit a closed–form solution, the non–negative nature of V and σ ensures a unique solution ξ_k for each t_k, which may be determined to high precision by standard numerical methods.

This philosophy distinguishes the present approach from that employed in [6], in which the differential equations characterizing the inverse dynamics scheme were formulated *ab initio* in terms of the curve parameter ξ, leading to more cumbersome analyses for the PI and PID controllers resulting from the third–order operator transformation from (15) to (16). A much simpler second–order transformation is employed in the present study, identified by the correspondence

$$L_t = a_2 D_t^2 + a_1 D_t + a_0 \;\rightarrow\; L_\xi = a_2 \left(\frac{V}{\sigma}\right)^2 D_\xi^2 + \left(a_2 \frac{\sigma V' - \sigma' V}{\sigma^2} + a_1\right) \frac{V}{\sigma} D_\xi + a_0 \qquad (19)$$

of linear differential operators in t and ξ.

6 Solution to inverse dynamics problem

The governing equations (5)–(7) have the general form (14), featuring a differential operator of higher order on the right than on the left. The counter–causal nature of these equations does not preclude the existence of a unique solution, for a given a set of initial conditions — the aim is to seek an *a priori* feed–forward modification of the input, for a known desired output, rather than finding the output for a given input, customarily appearing on the left and right sides of a differential equation. In essence, this is equivalent to a fundamental problem of classical mechanics — knowing the motion, determine the motive force.

The solution strategy may be summarized as follows.

1. By virtue of their LTI structure, the governing equations are solved in the time domain, with the required change of variables $t \rightarrow \xi$ performed only as a final step.

2. The initial conditions for the IDC correction $\Delta\mathbf{r}(t)$ at $t = 0$ are set to zero, to preclude additional terms dependent on the initial state. This amounts to specifying the order of kinematic continuity of $\Delta\mathbf{r}(t)$, or geometric contact between the modified and original paths $\hat{\mathbf{r}}(t)$ and $\mathbf{r}_d(t)$, at $t = 0$.

3. The linear equation $L_u\phi = g(u)$ can always be "reduced to integral form" — i.e., the solution $\phi(u)$ consists of closed–form terms and integrals involving the function $g(u)$. For LTI equations of the form (14), the goal of this Section is to reduce the solution to integrals involving only $x_d(t)$, with all its derivatives appearing outside these integrals.

As a consequence of the above approach, the general solution for the x–axis compensation $\Delta x(t)$ consists of three parts — differential, natural, and integral:

1. If L_t^c and L_t^d in (14) are of order m and n, the differential part includes derivatives of $x_d(t)$ up to order $n - m$.

2. The natural part is a linear combination of m characteristic exponential functions, with coefficients depending on the initial conditions for $x_d(t)$ and its derivatives up to order $n - 1$, at $t = 0$. It is one particular solution to the homogeneous equation $L_t^c \Delta x = 0$ associated with (14).

3. The integral part can be expressed as a single integral, whose integrand is the product of $x_d(t)$ with a linear combination of the m characteristic exponential functions. In the engineering context, such integrals are often represented in convolution form.

We now consider in more detail the specific forms of the solutions for the inverse dynamics compensation terms, in the context of P, PI, and PID controllers.

6.1 P controller

In the case of a P controller, L_t^c and L_t^d in (14) are of order 0 and 2, and equation (5) is itself the required solution for the compensation term,

$$\Delta x(t) = \frac{1}{K k_p}(J\ddot{x}_d + B\dot{x}_d),$$

in the time domain. It is essentially just an inversion of the axis motor equation $J\ddot{\theta} + B\dot{\theta} = T(t)$, adjusted for the plant and controller gains. Since (5) is not a differential equation for $\Delta x(t)$, the solution is unique and has only a differential part. No initial conditions can be set, and $\Delta x(t)$ is at least C^{-1} continuous, or has a jump discontinuity, at $t = 0$. Geometrically, the starting point of $\hat{\mathbf{r}}(t)$ does not coincide with that of $\mathbf{r}_d(t)$.

By the change of variables $t \to \xi$ using the corresponding operators (19), we obtain the solution expressed in terms of the curve parameter ξ, namely

$$\Delta x(\xi) = \frac{1}{K k_p}\left[J\left(\frac{V}{\sigma}\right)^2 x_d'' + \left(J\,\frac{\sigma V' - \sigma' V}{\sigma^2} + B\right)\frac{V}{\sigma}\,x_d'\right]. \tag{20}$$

To use the modified path $\hat{\mathbf{r}}(\xi)$ as input, real–time interpolation must be applied to (20), and analogous terms for the other axes, and the resulting path corrections $\Delta\mathbf{r}(\xi_k)$ must be added to the reference points $\mathbf{r}_d(\xi_k)$ on the original desired path.

6.2 PI controller

Setting $c = k_i/k_p$ and $L_t^d = JD_t^3 + BD_t^2$, the general solution to equation (6), written in the form

$$\Delta\dot{x} + c\,\Delta x = \frac{1}{K k_p} L_t^d x_d,$$

can be expressed as

$$\Delta x(t) = \mathrm{e}^{-ct}\left[\frac{1}{Kk_p}\int_0^t \mathrm{e}^{c\tau}\, L_\tau^d x_d(\tau)\,\mathrm{d}\tau + \Delta x(0)\right].$$

With initial condition $\Delta x(0) = 0$, this can be formulated as a single convolution integral

$$\Delta x(t) = \frac{1}{Kk_p}\int_0^t \mathrm{e}^{-c(t-\tau)}\, L_\tau^d x_d(\tau)\,\mathrm{d}\tau. \tag{21}$$

Using successive integration by parts, or by considering the polynomial division

$$\frac{L_\tau^d}{D_\tau + c} = L_\tau^q + \frac{r_0}{D_\tau + c}$$

of LTI differential operators, where

$$L_\tau^q = JD_\tau^2 - (Jc - B)D_\tau + c(Jc - B), \quad r_0 = -\,c^2(Jc - B), \tag{22}$$

or equivalently by noting that

$$\mathrm{e}^{-c(t-\tau)}(D_\tau + c)L_\tau^q x_d(\tau) = D_\tau[\,\mathrm{e}^{-c(t-\tau)}L_\tau^q x_d(\tau)\,],$$

the integral solution (21) can be further reduced as

$$\begin{aligned}\Delta x(t) &= \frac{1}{Kk_p}\left[\left[\,\mathrm{e}^{-c(t-\tau)}\, L_\tau^q x_d(\tau)\,\right]_{\tau=0}^{\tau=t} + r_0\int_0^t \mathrm{e}^{-c(t-\tau)}\, x_d(\tau)\,\mathrm{d}\tau\right]\\ &= \frac{1}{Kk_p}\left[L_t^q x_d(t) - L_t^q x_d(0)\,\mathrm{e}^{-ct} + r_0\int_0^t \mathrm{e}^{-c(t-\tau)}\, x_d(\tau)\,\mathrm{d}\tau\right].\end{aligned} \tag{23}$$

Note that each component (differential, natural, and integral) of the solution appear as the first, second, and third terms in (23). The natural part $-\,L_t^q x_d(0)\,\mathrm{e}^{-ct}$ serves as a "corrective" term, ensuring that $\Delta x(t)$ is at least C^0 continuous at $t = 0$, in agreement with the imposed initial condition $\Delta x(0) = 0$ when $\hat{\mathbf{r}}(t)$ starts from the same initial position as $\mathbf{r}_d(t)$.

To cast the solution (23) in terms of the curve parameter ξ instead of time t, we invoke the second–order operator transformation (19) to obtain

$$\Delta x(\xi) = \frac{1}{Kk_p}\left[L_\xi^q x_d(\xi) - L_\xi^q x_d(0)\,\mathrm{e}^{-ct(\xi)} + r_0\int_0^\xi \mathrm{e}^{-c\,[\,t(\xi)-t(\eta)\,]}\, x_d(\eta)\,\frac{\sigma(\eta)}{V(\eta)}\mathrm{d}\,\eta\right], \tag{24}$$

where

$$L_\xi^q = J\left(\frac{V}{\sigma}\right)^2 D_\xi^2 + \left[J\,\frac{\sigma V' - \sigma' V}{\sigma^2} - (Jc - B)\right]\frac{V}{\sigma}\,D_\xi + c(Jc - B), \quad r_0 = -\,c^2(Jc - B).$$

The term $L_t^q x_d(t)$ in (23) is evaluated using $L_\xi^q x_d(\xi)$ as specified above, with the parameter value ξ corresponding to time t obtained from the real–time interpolator. The integral term in (23) can be directly evaluated by numerical quadrature (see Section 8.1 below). Note that

the coefficient of the natural part can be denoted $L^q x_d(0)$, with an unspecified independent variable, since $L^q_t x_d(0) = L^q_\xi x_d(0) = L^q_s x_d(0)$ is the same constant.

Special case. If the ratio of the controller gains satisfies $c = k_i/k_p = B/J$, then (6) reduces to the explicit solution

$$\Delta x(t) = \frac{1}{Kl}\,\ddot{x}_d, \tag{25}$$

where $l = k_p/J = k_i/B$ (this can also be deduced from (23) by inspection). This special case corresponds to controller and plant transfer functions

$$G_c(s) = k_p + \frac{k_i}{s} = l\,\frac{Js+B}{s}, \quad G_p(s) = \frac{K}{s(Js+B)},$$

for which the machine axis dynamic impedance $Js+B$ cancels out in the product $G_c(s)G_p(s)$, yielding a double–integrator open–loop transfer function $G(s) = Kl/s^2$, equivalent to (25). With identical axis parameters, equation (25) specializes to the vector form

$$\Delta\mathbf{r}(t) = \frac{1}{Kl}\,\ddot{\mathbf{r}}_d.$$

Since $\ddot{\mathbf{r}}_d = \dot{V}\mathbf{t} + \kappa V^2\mathbf{n}$, where $\kappa, \mathbf{t}, \mathbf{n}$ are the curvature and unit tangent and normal of $\mathbf{r}_d(\xi)$, for a fixed feedrate $V = V_0$ the acceleration is in the curve normal direction, i.e.,

$$\Delta\mathbf{r}(t) = \frac{\mathbf{a}_n}{Kl} = \frac{\kappa V_0^2\mathbf{n}}{Kl}. \tag{26}$$

This is a purely geometrical relation, that holds whether one uses t, ξ, or s as the independent variable.

6.3 PID controller

Equation (7) can be analyzed by the methods used for the case of a PI controller. However, to better exploit the LTI structure of (7), we prefer to now use the Laplace transform and operator algebra. The Laplace transform of (7), with zero initial conditions for $\Delta x(t)$, yields

$$(s+r_1)(s+r_2)\Delta X(s) = \frac{1}{Kk_d}\,[\,(Js^3+Bs^2)X_d(s) - X_{d,0}(s)\,],$$

where we set

$$s^2 + \frac{k_p}{k_d}s + \frac{k_i}{k_d} = (s+r_1)(s+r_2), \tag{27}$$

and

$$X_{d,0}(s) = Jx_d(0)s^2 + [\,J\dot{x}_d(0) + Bx_d(0)\,]\,s + [\,J\ddot{x}_d(0) + B\dot{x}_d(0)\,].$$

Hence in the complex frequency domain, we obtain the rational solution

$$\Delta X(s) = \frac{1}{Kk_d}\,\frac{(Js^3+Bs^2)X_d(s) - X_{d,0}(s)}{(s+r_1)(s+r_2)}.$$

The Laplace variable s plays the same role as the derivative operator D_t. Depending on the nature of the roots of the characteristic polynomial (27), three cases may be identified.

1. $r_1 \neq r_2$ (distinct real roots). Polynomial division and partial fraction expansion leads to the frequency domain solution

$$\Delta X(s) = \frac{1}{Kk_d}\left[G^q(s)X_d(s) - Jx_d(0) + \frac{L_t^{q,1}x_d(0)}{s+r_1} - \frac{L_t^{q,2}x_d(0)}{s+r_2} + F(s)X_d(s) \right],$$

where

$$G^q(s) = Js - [\, J(r_1+r_2) - B\,],$$
$$L_t^{q,i} = \frac{1}{r_1-r_2}[\, JD_t^2 - (Jr_i - B)D_t + r_i(Jr_i - B)\,], \quad i = 1,2,$$
$$F(s) = \frac{1}{r_1-r_2}\left[r_1^2\,\frac{Jr_1-B}{s+r_1} - r_2^2\,\frac{Jr_2-B}{s+r_2} \right].$$

Transformed back to the time domain, the solution (containing the derivative, natural, and integral terms) reads

$$\Delta x(t) = \frac{1}{Kk_d}\left[L_t^q x_d(t) + L_t^{q,1}x_d(0)\,\mathrm{e}^{-r_1 t} - L_t^{q,2}x_d(0)\,\mathrm{e}^{-r_2 t} + \int_0^t f(t-\tau)\,x_d(\tau)\,\mathrm{d}\tau \right], \quad (28)$$

where

$$L_t^q = JD_t - [\, J(r_1+r_2) - B\,],$$
$$L_t^{q,i} = \frac{1}{r_1-r_2}[\, JD_t^2 - (Jr_i - B)D_t + r_i(Jr_i - B)\,], \quad i = 1,2,$$
$$f(t) = \frac{1}{r_1-r_2}\left[\, r_1^2(Jr_1-B)\,\mathrm{e}^{-r_1 t} - r_2^2(Jr_2-B)\,\mathrm{e}^{-r_2 t}\,\right]. \quad (29)$$

Observe that the coefficients $L_t^{q,i}x_d(0)$, $i = 1,2$ of the natural part are similar in form to the coefficient $L_t^q x_d(0)$ in the PI controller solution (23), involving all initial conditions of $x_d(t)$.

2. $r_1 = r_2 = r$ (a double root). By similar reasoning, we obtain the time solution

$$\Delta x(t) = \frac{1}{Kk_d}\left[L_t^{q,1}x_d(t) - \mathrm{e}^{-rt}\left[\, L_t^{q,1}x_d(0) + L_t^{q,2}x_d(0)\,t\,\right] + \int_0^t f(t-\tau)\,x_d(\tau)\,\mathrm{d}\tau \right],$$

where

$$L_t^{q,1} = JD_t - (2Jr - B), \quad L_t^{q,2} = JD_t^2 - (Jr-B)D_t + r(Jr-B),$$
$$f(t) = \mathrm{e}^{-rt}\left[\, r(3Jr-2B) - r^2(Jr-B)\,t\,\right].$$

3. $r_1, r_2 = \lambda \pm \mathrm{i}\,\mu$ (complex conjugate roots). The solution (28) also holds in this case, but may be re–written in the explicitly real form

$$\Delta x(t) = \frac{1}{Kk_d}\left[L_t^{q,a}x_d(t) - \mathrm{e}^{-\lambda t}\Big(L_t^{q,a}x_d(0)\cos\mu t + L_t^{q,b}x_d(0)\sin\mu t \Big) + \int_0^t f(t-\tau)\,x_d(\tau)\,\mathrm{d}\tau \right],$$

with

$$L_t^{q,a} = JD_t - (2J\lambda - B), \quad L_t^{q,b} = \frac{1}{\mu}[\, JD_t^2 - (J\lambda - B)D_t + J(\lambda^2 - \mu^2) - B\lambda\,],$$

$$f(t) = \mathrm{e}^{-\lambda t}(a\cos\mu t + b\sin\mu t), \tag{30}$$

where

$$a = J(3\lambda^2 - \mu^2) - 2B\lambda, \quad b = \frac{1}{\mu}\,[\, J\lambda(3\mu^2 - \lambda^2) + B(\lambda^2 - \mu^2)\,].$$

These results can also be expressed in terms of the quantities ω_n and ζ defined by $r_1, r_2 = \zeta\omega_n \pm \mathrm{i}\,\omega_n\sqrt{1-\zeta^2}$.

It can be verified that, in all the above cases, $\Delta x(t)$ is of (at least) C^1 continuity at $t = 0$, consistent with zero initial conditions for $\Delta x(t)$. The modified commanded path $\hat{\mathbf{r}}(t)$ is tangent to the original path $\mathbf{r}_d(t)$ at $t = 0$.

The solution $\Delta x(\xi)$, expressed in terms of the curve parameter, can be obtained by the same approach used in the case of a PI controller. For brevity, we omit the details here.

Special case. If $r_1 = B/J$ and r_2 is any real value γ, equation (7) can be written in the form

$$(JD_t + B)(D_t + \gamma)\Delta x = \frac{1}{Kl}(JD_t + B)D_t^2\, x_d,$$

where $l = k_d/J$. Cancelling the common factor $(JD_t + B)$, this is equivalent to

$$\Delta\dot{x} + \gamma\,\Delta x = \frac{1}{Kl}\,\ddot{x}_d, \tag{31}$$

which, for $\Delta x(0) = 0$, has the simple time solution

$$\Delta x(t) = \frac{1}{Kl}\left[\, \dot{x}_d(t) - \gamma\, x_d(t) - [\,\dot{x}_d(0) - \gamma\, x_d(0)\,]\,\mathrm{e}^{-\gamma\, t} + \gamma^2 \int_0^t \mathrm{e}^{-\gamma(t-\tau)}\, x_d(\tau)\,\mathrm{d}\tau \right].$$

Note that this solution also encompasses the case $r_1 = r_2 = B/J$.

6.4 Advantages of time-domain solution

To illustrate the advantage of solving for the path comepensation in terms of time t, rather than the path parameter ξ, consider the case of the PI controller discussed in Section 6.2. Equation (6), expressed in terms of ξ, has the form

$$\Delta x' + c\,\frac{\sigma}{V}\,\Delta x = \frac{1}{Kk_p}\frac{\sigma}{V}\,L_\xi^d x_d, \tag{32}$$

where L_ξ^d is the third–order differential operator (16), with non–constant coefficients defined by $(a_0, a_1, a_2, a_3) = (0, 0, B, J)$ in (17). The general solution of (32) has the integral form

$$\Delta x(\xi) = \exp\!\left(-\,c\int_0^\xi \frac{\sigma}{V}\,\mathrm{d}u\right)\left[\, \frac{1}{Kk_p}\int_0^\xi \exp\!\left(c\int_0^\eta \frac{\sigma}{V}\,\mathrm{d}u\right) L_\eta^d x_d\,\frac{\sigma}{V}\,\mathrm{d}\eta + \Delta x(0)\right],$$

and by choosing zero initial conditions and combining the exponentials, we obtain

$$\Delta x(\xi) = \frac{1}{Kk_p} \int_0^\xi \exp\left(- c \int_\eta^\xi \frac{\sigma}{V} \, \mathrm{d}u \right) L_\eta^d x_d \, \frac{\sigma}{V} \, \mathrm{d}\eta. \tag{33}$$

This is not a convolution in the strict sense, but may be considered a generalized convolution. It is the linear nature of the differential equation (32) that permits this compact integral expression for the correction term. However, unlike (21) for the time–domain solution, the integral (33) does not (in general) admit further simplification. Note also that the integrand depends on both $x_d(\xi)$ and its derivatives.

The typical case of a constant feedrate, $V = V_0$, is worth noting. In this case, the arc length is $s = V_0 t$, and the expression of the path correction in terms of s can be written as

$$\Delta x(s) = \frac{1}{Kk_p} \int_0^s \mathrm{e}^{-c(s-u)/V_0} L_u^d x_d(u) \, \frac{1}{V_0} \, \mathrm{d}u, \quad L_s^d = JV_0^3 D_s^3 + BV_0^2 D_s^2.$$

This can be further reduced to a solution in three parts,

$$\Delta x(s) = \frac{1}{Kk_p} \left[L_s^q x_d(s) - L_s^q x_d(0) \, \mathrm{e}^{-cs/V_0} + \frac{r_0}{V_0} \int_0^s \mathrm{e}^{-c(s-u)/V_0} \, x_d(u) \, \mathrm{d}u \right],$$

$$L_s^q = JV_0^2 D_s^2 - (Jc - B) V_0 D_s + c(Jc - B), \quad r_0 = -c^2 (Jc - B),$$

but this is essentially equivalent to the integral reduction in time (21)–(23), using $s = V_0 t$, and is valid only for a constant feedrate V_0 with corresponding coefficients $\beta_i = a_i V_0^i$ in (18).

As observed in Section 4, it is essential in real–time motion control at a prescribed speed V along a parametric curve $\mathbf{r}_d(\xi)$ to determine the relationship between the curve parameter ξ and elapsed time t, so as to compute a commanded machine position $\mathbf{r}_d(\xi_k)$ at each sampling time $t_k = k\Delta t$ — a task accomplished by the real–time interpolator function. The preceding analysis shows that, in the context of the inverse dynamics problem, formulating the path compensation term as a function $\Delta\mathbf{r}(t)$ of time (rather than the curve parameter ξ as in [6]) offers a much simpler expression and more precise computation of it. Specifically, for the PI and PID controllers, the path compensation can be reduced to closed–form terms and a single irreducible integral containing the the desired path $\mathbf{r}_d(\xi)$ but not its derivatives, which can be efficiently computed to machine precision by a simple numerical quaraduture scheme.

The equivalent formulation in terms of ξ, as employed in [6], involves integrals that are much more cumbersome and unwieldy — and consequently less amenable to accurate and efficient real–time evaluation. Obviously, time t is the natural independent variable in any dynamics problem, and formulating the inverse dynamics solution in terms of t, with a conversion to the corresponding parameter value ξ invoked only as a final step by the real–time interpolator, is evidently the most advantageous strategy.

7 Lead curve design

To smoothly execute the prescribed path $\mathbf{r}_d(\xi)$, with non–zero initial and final feedrates, it is necessary to incorporate "lead–in" and "lead–out" segments to allow for starting and stopping at rest.[4] Simple (linear or circular) segments are typically employed as the lead curve, which must itself be modified by an inverse dynamics compensation term, to ensure an appropriately smooth transition into and out of the curve $\mathbf{r}_d(\xi)$. For brevity, we discuss only the case of lead–in curves below — the case of lead–out curves is closely analogous.

As previously noted, if zero initial conditions for the differential equation (14) are assumed, the compensatory displacement $\Delta x(t)$ is at least C^{m-1} continuous at $t = 0$ with $\Delta x(t) \equiv 0$ for $t < 0$, where m is the order of the operator L_t^c ($m = 0, 1, 2$ for P, PI, PID controllers). This convention yields not only the formally simplest expression for $\Delta x(t)$ but also, when imposed on all axes, a modified path $\hat{\mathbf{r}}(t)$ that is geometrically closest to the prescribed path $\mathbf{r}_d(t)$ at $t = 0$. Conversely, to determine $\Delta x(t)$, one must specify $n - 1$ initial conditions for $x_d(t)$ at $t = 0^-$, where n is the order of L_t^d. At its juncture with $\mathbf{r}_d(t)$, the lead–in curve $\mathbf{p}_d(t)$ must match these initial conditions, ensuring a C^{n-1} connection and, as a necessary condition, G^{n-1} geometric continuity of $\mathbf{p}_d(t)$ with $\mathbf{r}_d(t)$ at $t = 0$.

We describe below the construction of linear and circular lead–in segments $\mathbf{p}_d(t)$ with C^1 and C^2 connections to the curve $\mathbf{r}_d(t)$ for P and PI/PID controllers, respectively. Note that the inverse dynamics scheme formulated for $\mathbf{r}_d(t)$ should also be applied to $\mathbf{p}_d(t)$ — the main difference is with the initial conditions: the lead–in motion starts from rest.

7.1 Lead-in linear segment for P controller

Assuming that traversal of the curve $\mathbf{r}_d(t)$ is to begin at time $t = 0$ with velocity $\mathbf{V}_0 = \dot{\mathbf{r}}_d(0)$, corresponding to feedrate $V_0 = |\mathbf{V}_0|$, we consider uniform acceleration from rest over a time interval T along a straight–line lead–in segment, defined by

$$\mathbf{p}_d(t) = \mathbf{r}_d(0) + \mathbf{V}_0 \, \frac{t(t+2T)}{2\,T}, \quad t \in [\,-T, 0\,].$$

The lead–in segment has start point $\mathbf{p}_d(-T) = \mathbf{r}_d(0) - \frac{1}{2}\mathbf{V}_0 T$ and length $L = \frac{1}{2}V_0 T$.

Substituting into (5) gives the lead–in segment path compensation term as

$$\Delta p_i(t) = \frac{1}{K_i k_{pi}} \left[J_i \frac{V_{0i}}{T} + B_i \frac{V_{0i}}{T} \, (t+T) \right],$$

where $i = x, y, z$ denotes a coordinate component (with possibly different gains and physical parameters for each axis) or, in vector notation,

$$\Delta \mathbf{p}(t) = \frac{\mathbf{J}}{T} + \frac{\mathbf{B}}{T} \, (t+T),$$

[4] When $\mathbf{r}_d(\xi)$ is an intermediate segment of a multi–segment path, these lead segments are only required for the initial and final segments.

where $\mathbf{J}$ and $\mathbf{B}$ have components $J_i V_{0i}/K_i k_{pi}$ and $B_i V_{0i}/K_i k_{pi}$. The sampling times $t_k = k\Delta t$ may be substituted directly into the modified lead–in path $\mathbf{p}_d(t) + \Delta\mathbf{p}(t)$ to obtain reference points for the lead–in motion.

Note that $\Delta\mathbf{p}(0) \neq \Delta\mathbf{r}(0)$ in general — although we impose $\dot{\mathbf{p}}_d(0) = \dot{\mathbf{r}}_d(0)$, the accelerations at the juncture point have not been matched, i.e., $\ddot{\mathbf{p}}_d(0) \neq \ddot{\mathbf{r}}_d(0)$. Thus, for a PI controller, the path correction term is generally C^{-1} (discontinuous) at the point $\mathbf{p}_d(0) = \mathbf{r}_d(0)$, even though $\mathbf{p}_d(t)$ meets $\mathbf{r}_d(t)$ with C^1 continuity at that point. The same is true for the juncture of the end point of $\mathbf{r}_d(t)$ with the lead–out segment.

Finally, we observe that the modified path

$$\hat{\mathbf{p}}(t) = \mathbf{p}_d(t) + \Delta\mathbf{p}(t) = \mathbf{r}_d(0) + \frac{1}{T}\left[\,\mathbf{J} + \mathbf{B}\,(t+T) + \tfrac{1}{2}\mathbf{V}_0\,t(t+2T)\,\right]$$

is, in general, a parabola segment. Under the assumption of identical axes parameters, the vectors $\mathbf{J}, \mathbf{B}, \mathbf{V}_0$ are paralell, and we obtain

$$\hat{\mathbf{p}}(t) = \mathbf{p}_d(t) + \Delta\mathbf{p}(t) = \mathbf{r}_d(0) + \frac{1}{T}\left[\,\frac{J + B(t+T)}{K k_p} + \tfrac{1}{2}\,t(t+2T)\,\right]\mathbf{V}_0,$$

which defines a constant–acceleration modified motion, in the direction of $\mathbf{V}_0$.

7.2 Lead-in circular arc for PI/PID controller

For the PI and PID controllers, a C^2 junction of the lead–in curve $\mathbf{p}_d(t)$, $t \in [-T, 0\,]$ with the desired path $\mathbf{r}_d(t)$ at $t = 0$ may be constructed. A circular arc in the *osculating plane* at $t = 0$ — i.e., the plane spanned by the initial tangent $\mathbf{t}_0$ and principal normal $\mathbf{n}_0$ — may be used for this purpose. We express this arc in terms of an angular function $\theta(t)$, with initial and final values $\theta(-T) = \theta_i$ and $\theta(0) = \theta_f$, so that

$$\mathbf{p}_d(t) = \mathbf{r}_d(0) + R\,[\,(\cos\theta(t) - \cos\theta_f)\,\mathbf{t}_0 + (\sin\theta(t) - \sin\theta_f)\,\mathbf{n}_0\,], \tag{34}$$

Taking the first and second derivatives of (34) yields

$$\begin{aligned}
\dot{\mathbf{p}}_d(t) &= R\,[\,\cos\theta(t)\,\mathbf{n}_0 - \sin\theta(t)\,\mathbf{t}_0\,]\,\dot{\theta}(t),\\
\ddot{\mathbf{p}}_d(t) &= R\,[\,\cos\theta(t)\,\mathbf{n}_0 - \sin\theta(t)\,\mathbf{t}_0\,]\,\ddot{\theta}(t) - R\,[\,\cos\theta(t)\,\mathbf{t}_0 + \sin\theta(t)\,\mathbf{n}_0\,]\,\dot{\theta}^2(t).
\end{aligned}$$

For a general feedrate $V(t)$ along $\mathbf{r}_d(t)$, the initial velocity and acceleration are $\mathbf{V}_0 = V_0\,\mathbf{t}_0$ and $\mathbf{A}_0 = \dot{V}_0\,\mathbf{t}_0 + \kappa_0 V_0^2\,\mathbf{n}_0$, where $V_0 = V(0)$, $\dot{V}_0 = \dot{V}(0)$, and κ_0 is the initial curvature. With $\dot{\theta}_f = \dot{\theta}(0)$ and $\ddot{\theta}_f = \ddot{\theta}(0)$, the C^2 matching conditions $\dot{\mathbf{p}}_d(0) = \mathbf{V}_0$ and $\ddot{\mathbf{p}}_d(0) = \mathbf{A}_0$ become

$$\begin{aligned}
R\,[\,\cos\theta_f\,\mathbf{n}_0 - \sin\theta_f\,\mathbf{t}_0\,]\,\dot{\theta}_f &= V_0\,\mathbf{t}_0,\\
R\,[\,\cos\theta_f\,\mathbf{n}_0 - \sin\theta_f\,\mathbf{t}_0\,]\,\ddot{\theta}_f - R\,[\,\cos\theta_f\,\mathbf{t}_0 + \sin\theta_f\,\mathbf{n}_0\,]\,\dot{\theta}_f^2 &= \dot{V}_0\,\mathbf{t}_0 + \kappa_0 V_0^2\,\mathbf{n}_0.
\end{aligned}$$

The first equation implies that $\cos\theta_f = 0$, $\sin\theta_f = \pm 1$, and taking $\theta_f = 3\pi/2$ yields $\dot{\theta}_f = V_0/R$. Substituting these values into the second equation then gives $R = 1/\kappa_0$ and $\ddot{\theta}_f = \dot{V}_0/R$. In

addition, to start from rest on the lead–in curve, we must have $\dot{\theta}_i = \dot{\theta}(-T) = 0$. The required values of θ_f, $\dot{\theta}_f$, $\ddot{\theta}_f$ and $\dot{\theta}_i$ are matched by the cubic polynomial

$$\theta(t) = \frac{3\pi}{2} + \kappa_0 V_0\, t + \frac{\kappa_0 \dot{V}_0}{2}\, t^2 + \frac{\kappa_0 (\dot{V}_0 T - V_0)}{3T^2}\, t^3, \quad t \in [-T, 0\,], \tag{35}$$

and the initial angle is then $\theta_i = \theta(-T) = 3\pi/2 - 2\,\kappa_0 V_0 T/3 + \kappa_0 \dot{V}_0 T^2/6$. From (35) one can deduce that $\dot{\theta}(t) = \kappa_0\,[\,V_0 + (\dot{V}_0 T - V_0)\,t/T\,]\,(1 + t/T)$, and hence $\theta(t)$ has a monotone variation for $t \in [\,-T, 0\,]$ when $2V_0 - \dot{V}_0 T > 0$. Assuming this condition holds, the angular extent $\Delta\theta = \theta_f - \theta_i$ and length $L = R\,\Delta\theta$ of the lead–in arc are

$$\Delta\theta = \frac{\kappa_0 (4V_0 - \dot{V}_0 T) T}{6} \quad \text{and} \quad L = \frac{(4V_0 - \dot{V}_0 T) T}{6}.$$

For a constant feedrate ($\dot{V}_0 = 0$) on $\mathbf{r}_d(t)$, we have $L = 2\,V_0 T/3$ and require $T < 3\pi |\,R\,|/V_0$ to ensure that $\Delta\theta < 2\,\pi$. For $\Delta\theta < 2\,\pi$ when $\dot{V}_0 \neq 0$, the duration T should satisfy

$$\dot{V}_0 T^2 - 4\,V_0 T + 12\pi R > 0.$$

From the form (34) for the circular lead–in arc, together with the angular function (35) and θ_i, θ_f as defined above, one can compute the IDC path modification term for the lead–in arc $\mathbf{p}_d(t)$ (assuming identical axes) by appropriate modifications of equation (23) to obtain

$$\Delta\mathbf{p}(t) = \frac{1}{K k_p} \left[L_t^q\, \mathbf{p}_d(t) - L_t^q\, \mathbf{p}_d(-T)\, \mathrm{e}^{-c(t+T)} + r_0 \int_{-T}^{t} \mathrm{e}^{-c(t+T-\tau)}\, \mathbf{p}_d(\tau)\, \mathrm{d}\tau \right],$$

where L_t^q and r_0 are defined by (22). This satisfies the initial condition $\Delta\mathbf{p}(-T) = \mathbf{0}$, but it is not possible to also make it satisfy $\Delta\mathbf{p}(0) = \mathbf{0}$, so there is (in general) a discontinuity in the correction term at the C^2 juncture $\mathbf{p}_d(0) = \mathbf{r}_d(0)$ of the lead–in arc and the given curve. Reference points along the modified lead–in arc are obtained by substituting $t_k = k\Delta t$ into (35), using the resulting $\theta(t_k)$ values in (34), and finally adding the correction term $\Delta\mathbf{p}(t_k)$.

8 Simulation results

The focus of this study is on developing the theoretical foundations for an inverse dynamics compensation scheme, in which computation of the path correction term is formulated in the time domain, and the corresponding curve parameter value is computed as a final step by the real–time interpolator algorithm. In a follow–up study, the IDC scheme will be implemented on an 3–axis CNC milling machine with an open–architecture controller, and experimental performance results from this implementation will be described and analyzed.

At present, we illustrate some representative results from numerical simulations of the IDC path modification scheme, based on the following assumptions.

1. Path geometry and kinematics: the desired motion $\mathbf{r}_d(t)$ is specified by a fixed feedrate V_0 along a PH curve $\mathbf{r}_d(\xi)$, $\xi \in [\,0, 1\,]$ in the (x, y) plane.

2. System parameters: the machine x and y axes are assumed to have identical physical and control parameters. This facilitates a simple vector formulation, and yields physical insight by analogy with the dynamics of a point mass.

We adopt the same settings as in [6] for the simulations. Figure 2 illustrates the test curve, a planar PH quintic of total arc length $S = 1.108$ m, with extreme parametric speed and curvature variations, to be traversed at the constant feedrate $V_0 = 0.12$ m/s. The curve is plotted with uniform parameter increments $\Delta\xi = 0.01$ and arc length increments $\Delta s = 0.01\, S$ (corresponding to a constant feedrate). The total traversal time is $T = 9.234$ s (regardless of whether the input is the original path or the IDC modified path). The physical parameters for both axes are $K = 0.008$ N $\cdot$ m, $J = 0.01$ kg $\cdot$ m^2, and $B = 0.025$ kg $\cdot$ m^2/s.

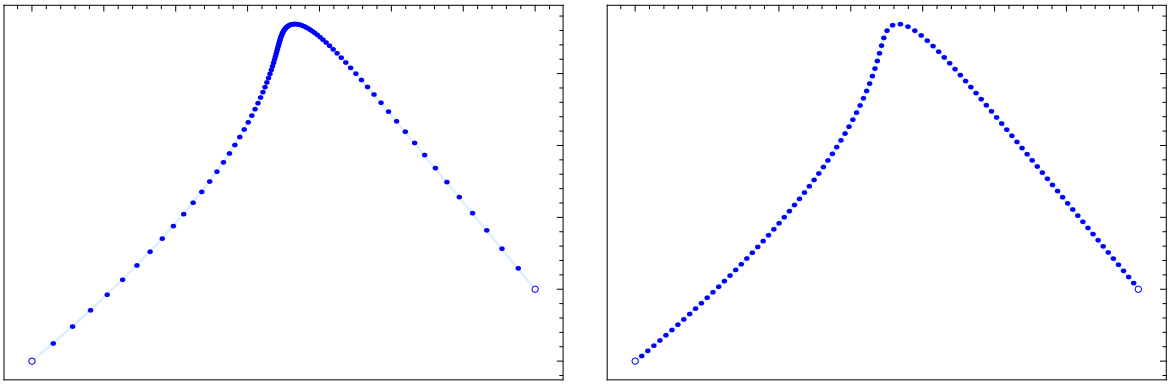

Figure 2: The PH quintic test curve $\mathbf{r}_d(\xi)$ rendered with uniform parameter increments $\Delta\xi$ (left), and uniform arc–length increments Δs that correspond to a constant feedrate (right).

In each of the following examples, a test case is specified by a set of PID control parameters.[5] We simulate the system response governed by the closed–loop transfer function (2), using both the original desired path $\mathbf{r}_d(t)$ and the IDC modified path $\hat{\mathbf{r}}(t)$ as input, and assess the performance in terms of transient response, steady–state behavior, and stability.

8.1 Numerical methods

For preliminary verification purposes, the simulations are based on the governing differential equations corresponding to (2) using the original desired path $\mathbf{r}_d(t)$ and the IDC modified path $\hat{\mathbf{r}}(t)$ as inputs, namely

$$J\,\dddot{\mathbf{r}}_o + B\,\ddot{\mathbf{r}}_o + K(k_d\,\ddot{\mathbf{r}}_o + k_p\,\dot{\mathbf{r}}_o + k_i\,\mathbf{r}_o) = K(k_d\,\ddot{\mathbf{r}}_d + k_p\,\dot{\mathbf{r}}_d + k_i\,\mathbf{r}_d), \tag{36}$$

$$J\,\dddot{\mathbf{r}}_d + B\,\ddot{\mathbf{r}}_d + K(k_d\,\ddot{\mathbf{r}}_d + k_p\,\dot{\mathbf{r}}_d + k_i\,\mathbf{r}_d) = K(k_d\,\ddot{\hat{\mathbf{r}}} + k_p\,\dot{\hat{\mathbf{r}}} + k_i\,\hat{\mathbf{r}}). \tag{37}$$

Here $\mathbf{r}_o(t)$ in (36) is the output path obtained without using the IDC scheme, whereas in (37) the IDC modified path is employed as the input, and the desired path $\mathbf{r}_d(t)$ is the expected

[5] Low gain values k_p, k_i, k_d are used here, to clearly illustrate the difference between the desired path $\mathbf{r}_d(\xi)$ and the modified path $\hat{\mathbf{r}}(\xi)$ — in practice, this difference will be less pronounced.

output (this output may exhibit minor deviations from $\mathbf{r}_d(t)$ due to numerical inaccuracies in the simulations). The numerical integration of these equations in the simulations was based on an order–adaptive predictor–corrector Adams method, yielding results of high accuracy.

The desired path $\mathbf{r}_d(t)$ and its derivatives can be evaluated by closed–form expressions. The modified input path $\hat{\mathbf{r}}(t)$ and its derivatives can be partly evaluated in the same manner, but the irreducible integral term requires a numerical integration. In PI control, for example, we compute the integral

$$\int_0^t \mathrm{e}^{-c(t-\tau)}\, \mathbf{r}_d(\xi(\tau))\, \mathrm{d}\tau \tag{38}$$

in (23) using a closed Newton–Cotes quadrature[6] [18] with m nodes $t_i = t_{k-1} + i\,\Delta t/(m-1)$, $i = 0, \ldots, m-1$ over the k^{th} sampling interval Δt, with the corresponding parameter values ξ_i required to evaluate the points $\mathbf{r}_d(\xi_i)$, $i = 0, \ldots, m-1$ being computed by the real–time interpolator. These integrals are accumulated over successive sampling intervals to obtain the total integral (38) over $[\,0, t\,]$. The integral over one sampling interval converges to double–precision accuracy using just $m = 4$ nodes. As another accuracy check, the integral (38) was also expressed in the form $\mathrm{e}^{-ct}\,\mathbf{u}(t)$, where $\mathbf{u}(t)$ satisfies the simple differential equation

$$\dot{\mathbf{u}}(t) = \mathrm{e}^{ct}\, \mathbf{r}_d(\xi(t)), \quad \mathbf{u}(0) = \mathbf{0}, \tag{39}$$

and $\xi(t)$ is approximated by a cubic spline interpolant. Numerical integration of (39) by the adaptive Adams method yields results in close agreement with the quadrature method.

The real–time interpolator is modelled as a zero order sample–hold with nominal frequency $f = 1\,\mathrm{kHz}$, for functions of both ξ and t, applied to the entire right–hand side of (36) and (37), which correspond to a "staircase" waveform input. The initial state of $\mathbf{r}_d(t)$ is imposed on both (36) and (37). In the latter case, the prescribed initial conditions obviate the need for a lead–in curve.

8.2 P controller

The proportional gain $k_p = 10$ yields a stable second–order control system with closed–loop poles at $s_{1,2} = -1.250 \pm 2.537\,\mathrm{i}$, corresponding to the constants

$$\text{exponential decay time } T_c = 0.8\text{ s}, \quad \text{oscillation period } T_p = 2.476\text{ s}, \tag{40}$$

which are both significantly shorter than the total traversal time $T = 9.234\,\mathrm{s}$, characterizing the fast and well–damped transient behavior of this good design.

Figure 3 compares simulation results using the original commanded path and IDC modified path as input. In the former case, an expected fast–decaying oscillation (of small magnitude) in the output can be observed, in response to the "sharp turn" of the curve. In the latter case, the pronounced looping of the IDC modified path "tricks" the system into producing the desired output path. This looping behavior corresponds to sudden commanded oscillations

[6] This quadrature has the advantage of positive rational weights at equisdistant nodes. Parameter values corresponding to nodes within a sampling interval are determined by the real–time interpolator.

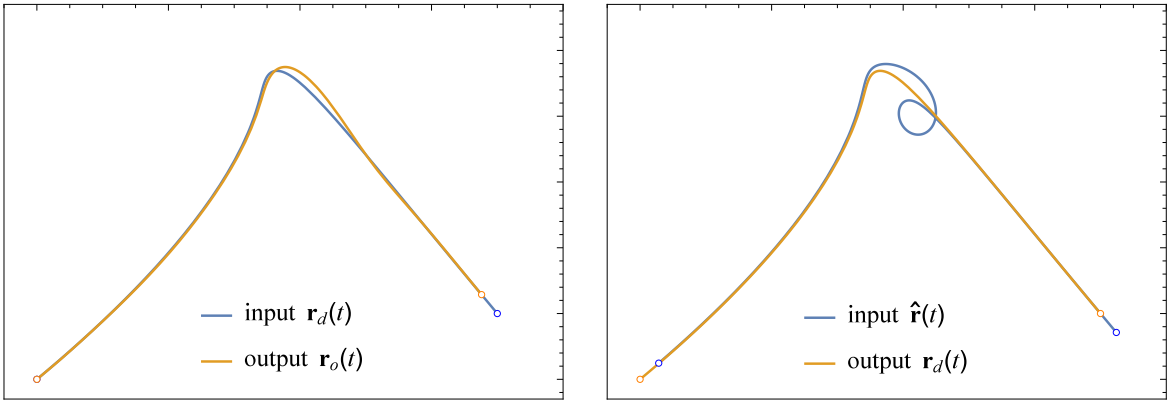

Figure 3: Simulation results for the P controller. Left: the output path $\mathbf{r}_o(t)$ obtained when the input is specified by the original desired path $\mathbf{r}_d(t)$. Right: the compensated output path $\mathbf{r}_d(t)$ obtained when the input is specified by the IDC modified path $\hat{\mathbf{r}}(t)$.

of the individual machine axes, whose effect is to produce a motion that accurately tracks the region of extreme curvature.

Note that the open–loop transfer function (3) for the P controller has a simple pole at origin (since $k_i = k_d = 0$), resulting in a finite steady–state tracking error for a ramp input. Since specifying either $\mathbf{r}_d(t)$ or $\hat{\mathbf{r}}(t)$ as the input path corresponds to an almost constant *velocity* along the two near–linear "legs" before and after the sharp turn, the theoretical value

$$e(\infty) = \frac{BV_0}{Kk_p} = 37.5\,\text{mm}$$

is a good estimate of position error along these legs — in the case of identical axis properties, this is predominantly feed error (i.e., lag).

Notwithstanding the accurate initial state, $\mathbf{r}_o(t)$ develops a delay if $\mathbf{r}_d(t)$ is used as input, with a final position error $|\mathbf{r}_o(T) - \mathbf{r}_d(T)| = 37.38$ mm, within 0.3 % of $e(\infty)$. By comparison, $\hat{\mathbf{r}}(t)$ is globally shifted ahead of $\mathbf{r}_d(t)$ by a finite lead (corresponding to the imposed C^{-1} continuity) which compensates for the steady–state error.

8.3 PI controller

The choices $k_p = 10$, $k_i = 10$ yields a stable system, with closed–loop poles $s_{1,2} = -0.629 \pm 2.458\,\text{i}$ and $s_3 = -1.243$. The complex conjugate poles determine the major oscillation mode, the values of the constants (40) being

$$T_c = 1.591\,\text{s} \quad \text{and} \quad T_p = 2.556\,\text{s}.$$

The real pole, with a time constant $\tau_c = 0.805$ s also contributes to the output. In this case, the system incurs no steady–state error in tracking a uniform motion.

Compared to the P controller, the simulation result for the original commanded path shown in Figure 4 exhibits an expected slower decay, a greater overshoot, and a similar frequency

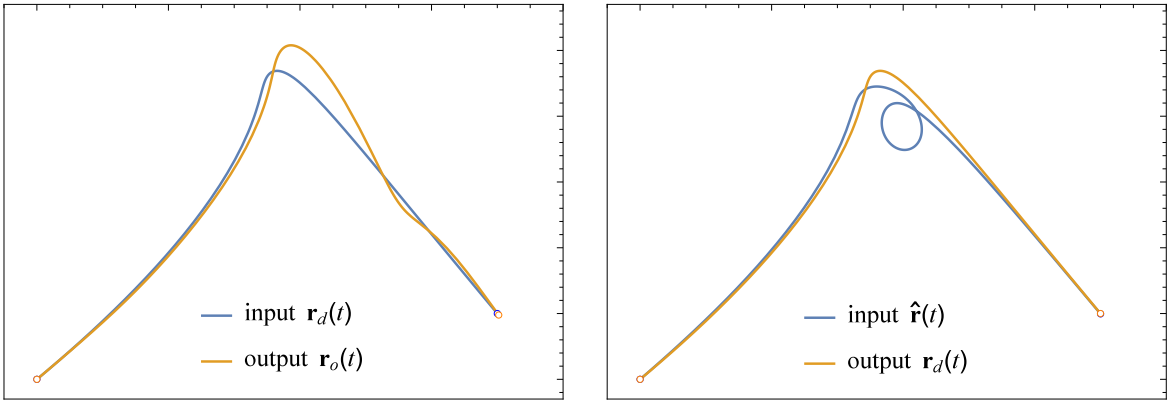

Figure 4: Simulation results for the PI controller. Left: the output path $\mathbf{r}_o(t)$ obtained when the input is specified by the original desired path $\mathbf{r}_d(t)$. Right: the compensated output path $\mathbf{r}_d(t)$ obtained when the input is specified by the IDC modified path $\hat{\mathbf{r}}(t)$.

of the curvature–induced oscillations, as well as a minimal tracking error at steady state (the superposed effect of the exponential mode with time constant $\tau_c = 0.805$ s is also discernible). Also shown in Figure 4 is the simulation result employing the IDC modified path as input — in this case, the commanded and executed motions start in unison and end nearly coincident, reflecting an almost–synchronized motion (as also evidenced by lower location of the loop in the modified path, compared to that in Figure 3).

For the IDC scheme in [6], formulated in terms of the curve parameter ξ rather than time t, a closed–form expression for the modified path $\hat{\mathbf{r}}(\xi)$ is not possible in the case of PI control, and an optimized polynomial approximation was employed. The present time–domain approach, in which $\hat{\mathbf{r}}(t)$ is evaluated from the analytic solution (23), is superior and yields results with numerical accuracy comparable to that in P control (for which a closed–form solution exists).

Special case. The special case (25) specifies a *marginally* stable closed–loop system, with poles $s_{1,2} = \pm\sqrt{Kl}\,\mathrm{i}$ that incur undamped oscillations in the output $\mathbf{r}_o(t)$ for the unmodified input $\mathbf{r}_d(t)$, and the output for the IDC modified input $\hat{\mathbf{r}}(t)$ is susceptible to disturbances.

The loop of $\hat{\mathbf{r}}(\xi)$ in Figure 4 is incurred by the need to accurately negotiate the high–curvature region of $\mathbf{r}_d(\xi)$, as indicated by the correction term (26). For the fixed feedrate $V_0 = 0.12$ m/s, the loop size is locally proportional to the curvature, and inversely proportional to the k_p gain value. Figure 5 shows the severe variation of curvature with arc length along $\mathbf{r}_d(\xi)$, and the pronounced effect of successively reducing k_p from 10 to 5 and 2.

8.4 PID controller

With three adjustable control gains, much greater freedom in the output behavior is possible, and greater care in their selection is necessary to ensure acceptable performance.

Example 1. With $k_p = 6$, $k_i = 25$, $k_d = 1$, the closed–loop transfer function has poles $s_{1,2} = 0.120 \pm 2.374\,\mathrm{i}$ and $s_3 = -3.540$, corresponding to a slightly unstable system, dominated by

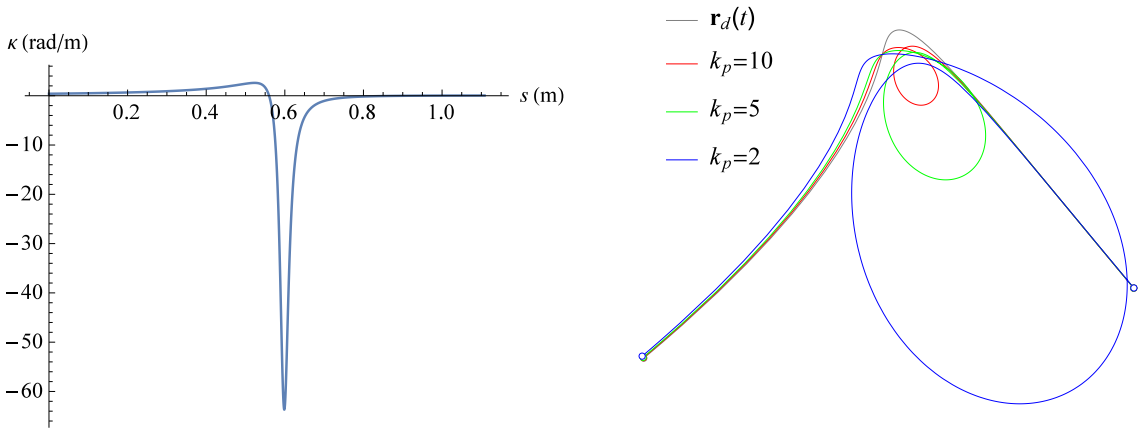

Figure 5: Left: curvature of the commanded path $\mathbf{r}_d(\xi)$ as a function of the path arc length. Right: the variation in size of the loop in the IDC modified path $\hat{\mathbf{r}}(\xi)$, for the special case (25) of the PI controller corresponding to $k_i/k_p = B/J$, using the gain values $k_p = 10, 5, 2$.

a slowly–divergent oscillation, as identified by the values

$$T_c = -\,8.333 \text{ s} \quad \text{and} \quad T_p = 2.647 \text{ s}$$

of the constants (40). There is no steady–state tracking error for a constant input velocity, but only in a *time–averaged* (i.e., oscillatory) sense.

The system response seen in Figure 6 has features in common with the PI controller, and a similar analysis applies. Surprisingly, however, an acute cusp–like feature[7] appears in the output $\mathbf{r}_o(t)$ after the sharp turn when using the original commanded path $\mathbf{r}_d(t)$ as input. Figure 6 shows a uniform time sampling of $\mathbf{r}_o(t)$, indicating that the motion nearly stops and then slowly reverses at this feature. When the IDC modified path $\hat{\mathbf{r}}(t)$ is used as input, the output adheres more closely to the desired path $\mathbf{r}_d(t)$, although some oscillation about it remains after the sharp turn. Figure 6 also shows a uniform time sampling of the modified input path $\hat{\mathbf{r}}(t)$, indicating a rapid commanded motion in the high curvature region of $\mathbf{r}_d(t)$.

Figure 7 shows the total and normal position errors, and axis accelerations, for the output motions. When the original path $\mathbf{r}_d(t)$ is used as input, pronounced oscillation of the position error is induced in the output motion after encountering the sharp turn (at $t \sim 5$ s). Similar oscillations of position error are observed when the IDC modified path $\hat{\mathbf{r}}(t)$ is used as input, but they are $\sim 10\times$ smaller (these oscillations are caused by small numerical inaccuracies, and reflect the marginally stable nature of the system). Note that, using $\mathbf{r}_d(t)$ as input, large axis acceleration oscillations develop on encountering the sharp turn, and persist thereafter. When the modified path $\hat{\mathbf{r}}(t)$ is used as input, large output axis accelerations occur mainly at the sharp turn, but persistent oscillations of lower magnitude are still discernible thereafter.

Example 2. In this case we use gains $k_p = 1$, $k_i = 1$, $k_d = 1$, which result in closed–loop poles $s_{1,2} = -0.0870 \pm 0.498\,\mathrm{i}$ and $s_3 = -3.126$. The complex–conjugate pole pair dominates

[7] Close inspection reveals that the reversals of the x and y axis motions occur only 0.04 s apart.

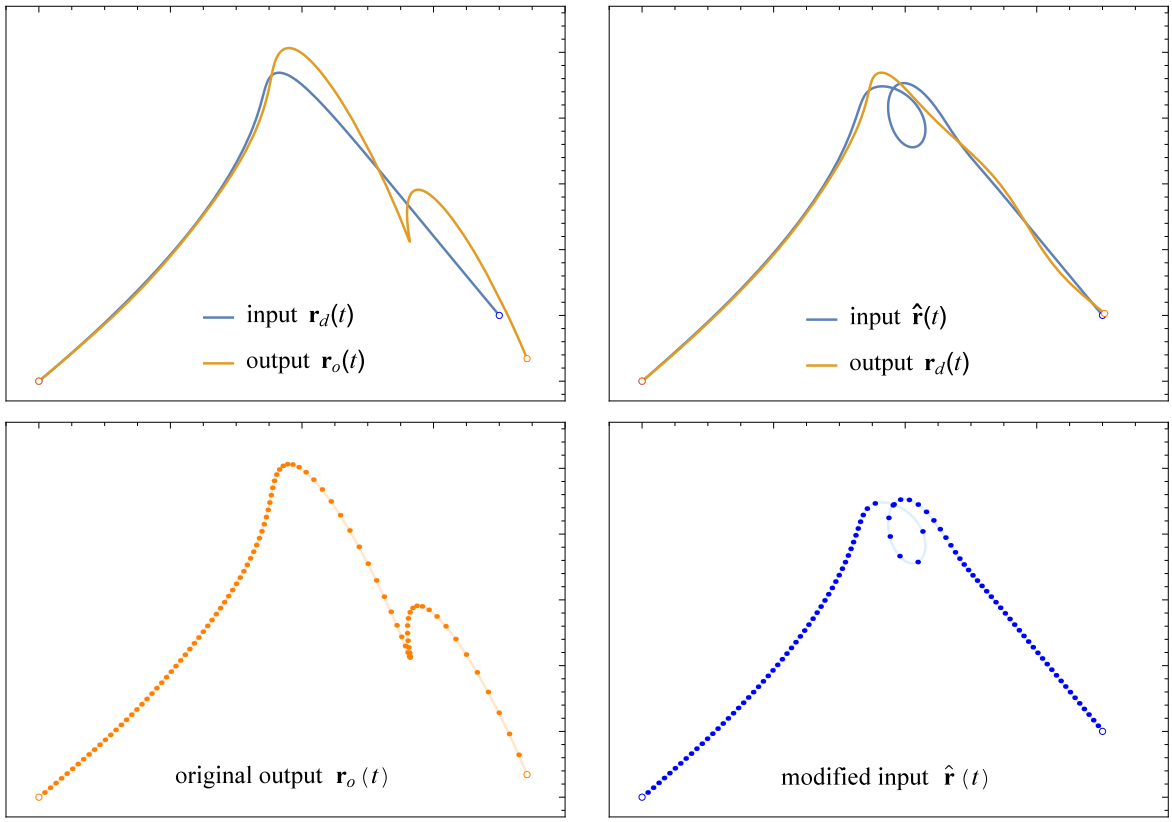

Figure 6: Simulation results for a PID controller in Example 1. Upper: the output path $\mathbf{r}_o(t)$ obtained (left) with input specified by the desired path $\mathbf{r}_d(t)$; and (right) the compensated output $\mathbf{r}_d(t)$ obtained with input specified by the IDC modified path $\hat{\mathbf{r}}(t)$. Lower: uniform time–sampling of (left) the original output $\mathbf{r}_o(t)$; and (right) the modified input $\hat{\mathbf{r}}(t)$.

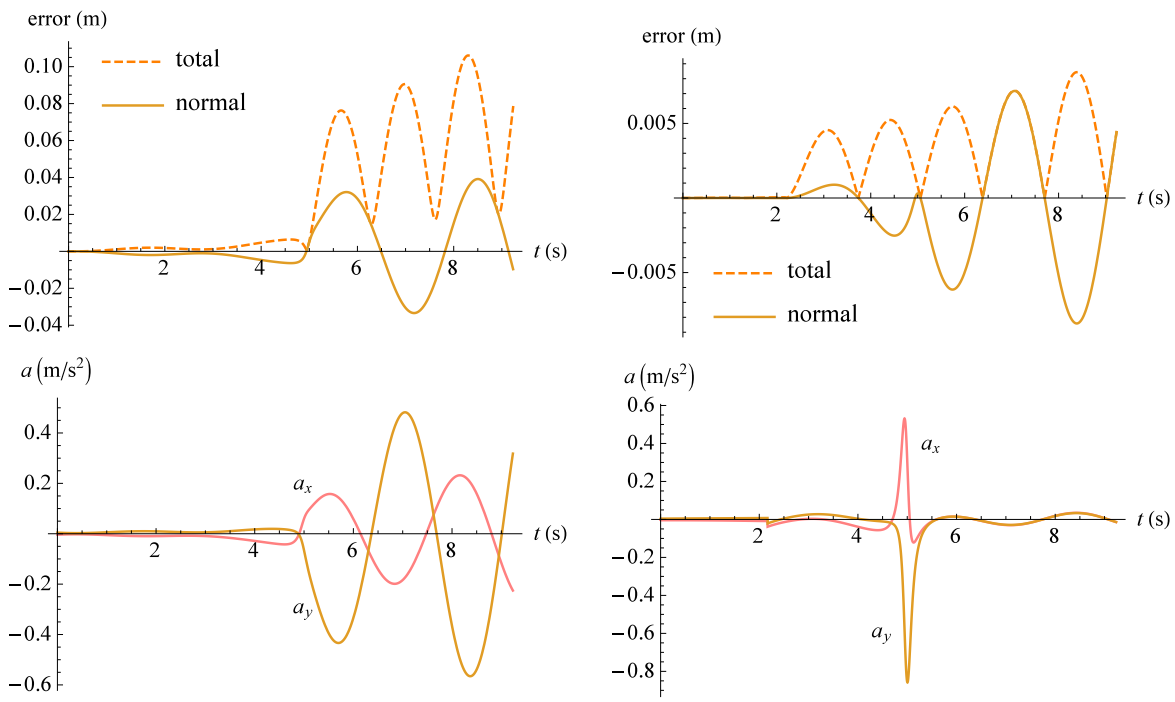

Figure 7: Upper: total and normal position errors along the output path in Example 1 with a PID controller, using the original desired path (left) and IDC modified path (right) as input. Lower: axis accelerations in these two cases. Note the different vertical scales in these plots.

this barely–stable system (similar to the influence of a pole at the origin), and the values of the constants (40) are

$$T_c = 11.49 \text{ s} \quad \text{and} \quad T_p = 12.61 \text{ s}.$$

Thus, one expects a persistent transient phase (T_c) with long–period oscillations (T_p), but the output should eventually converge accurately for a uniform input motion.

This behavior is confirmed in the simulation results illustrated in Figure 8. The output $\mathbf{r}_o(t)$ exhibits substantial deviation from the original path $\mathbf{r}_d(t)$ specified as input, even along the gentle left leg of the path, which was accurately tracked in all the preceding examples. This tracking error persists and grows, resulting in an "outrageous" ultimate position error. On the other hand, when the IDC modified path $\hat{\mathbf{r}}(t)$ — which does not loop, but nevertheless obviously differs substantially from $\mathbf{r}_d(t)$ — is employed as the input, the large compensation results in a remarkably accurate tracking of the original desired path.

Figure 8 also shows uniform time–samplings of the output $\mathbf{r}_o(t)$ when $\mathbf{r}_d(t)$ is used as input, and of the IDC modified input path $\hat{\mathbf{r}}(t)$. In the former case, the output is seen to be quite smooth, although wildly inaccurate. In the latter case, the remarkable non–uniformity of the commanded positions along the modified path reflects the extraordinary measures exerted by the IDC scheme to ensure accurate tracking of the sharp turn in the desired path.

Further details on this case can be found in the position error and axis acceleration plots in Figure 9. The following points are noteworthy.

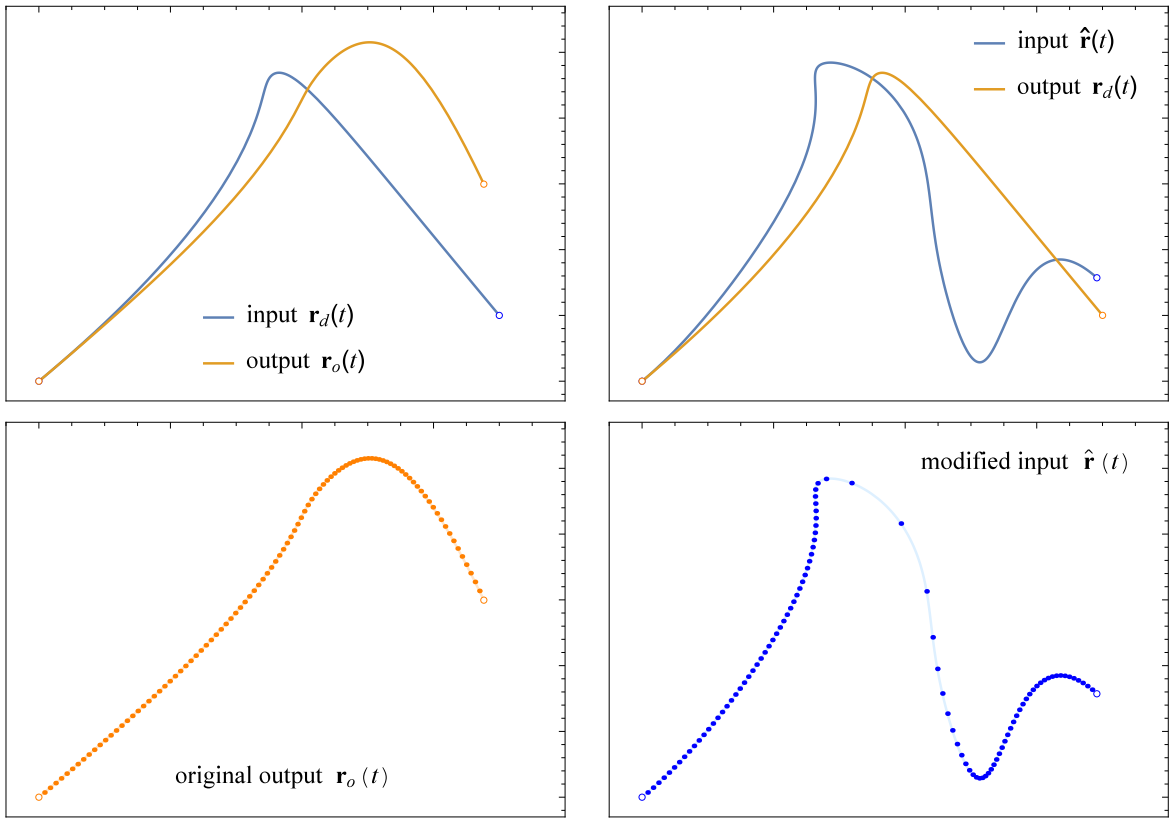

Figure 8: Simulation results for a PID controller in Example 2. Upper: the output path $\mathbf{r}_o(t)$ obtained (left) with input specified by the desired path $\mathbf{r}_d(t)$; and (right) the compensated output $\mathbf{r}_d(t)$ obtained with input specified by the IDC modified path $\hat{\mathbf{r}}(t)$. Lower: uniform time–sampling of (left) the original output $\mathbf{r}_o(t)$; and (right) the modified input $\hat{\mathbf{r}}(t)$.

1. Continuity: $\mathbf{r}_o(t)$ has G^2 contact with $\mathbf{r}_d(t)$ and $\hat{\mathbf{r}}(t)$ has G^1 contact with $\mathbf{r}_d(t)$ at $t = 0$ — in accordance with the imposed C^2 and C^1 kinematic continuity conditions.

2. Final state: both the original output $\mathbf{r}_o(t)$ and the IDC modified input $\hat{\mathbf{r}}(t)$ approach uniform motion along the right leg of $\mathbf{r}_d(t)$, but do not quite settle down by the end of the run. Note that the end state of $\hat{\mathbf{r}}(t)$ differs in each individual case, and in general does not coincide with that of $\mathbf{r}_d(t)$.

3. Acceleration: the peak output axis accelerations observed when using $\hat{\mathbf{r}}(t)$ as the input are substantially higher, but of shorter duration (arising from negotiation of the sharp turn), compared to those using $\mathbf{r}_d(t)$ as input. Note, however, that the goal of the IDC compensation scheme is to ensure *positional* accuracy, and this may necessitate large axis accelerations (which should not exceed the drive motor limits).

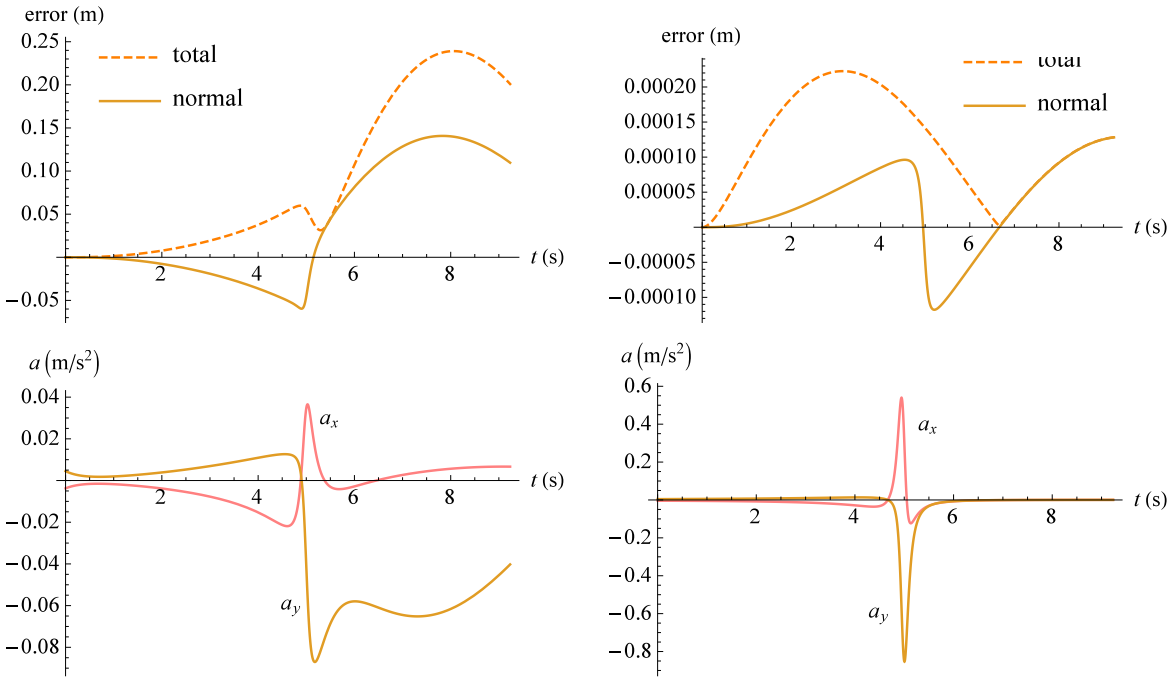

Figure 9: Upper: total and normal position errors along the output path in Example 2 with a PID controller, using the original desired path (left) and IDC modified path (right) as input. Lower: axis accelerations in these two cases. Note the different vertical scales in these plots.

Special case. Unlike equation (25) in the case of PI control, equation (31) for PID control defines a stable second–order system with two parameters l, c which (with J and B cancelled out) may be freely selected. This case may, in principle, mimic any of the preceding stable examples that converge to constant velocity.

9 Closure

In the formulation of an inverse dynamics compensation (IDC) scheme for real–time motion control with constant or variable speed along a parametric curve $\mathbf{r}_d(\xi)$, the control algorithm

must determine a curve parameter value ξ_k corresponding to each sampling time $t_k = k\Delta t$, allowing a reference point (commanded position) to be computed. When the IDC differential equations are cast in terms of ξ, however, they acquire non–constant coefficients that preclude a closed–form solution for the path correction, except in the simple case of a P controller [6].

The principal contribution of this study has been to formulate analytic solutions for the IDC path correction term in the case of PI and PID controllers, based on the time–domain IDC differential equations with constant coefficients. Analytic expressions for the path correction were derived, in terms of the commanded path $\mathbf{r}_d(\xi)$ and its derivatives, and an irreducible integral (which can be computed to machine precision in each sampling interval by a simple quadrature), dependent on $\mathbf{r}_d(\xi)$ but not its derivatives. The parameter value ξ_k required to obtain the reference point $\mathbf{r}_d(\xi_k)$ at each sampling time $t_k = k\Delta t$ is computed as a final step, by the real–time interpolator. This simplifies computation of the IDC correction, and greatly improves its accuracy and efficiency. The conceptual framework and mathematical analysis underlying the extension of the IDC scheme to PI and PID controllers was developed in detail, and illustrated using simulations. The design of lead–in/out segments, facilitating motion along a curved path with smooth transitions from and to rest, was also developed. For a given controller, the IDC scheme is expected to yield significant improvements in the tracking accuracy of strongly–curved paths traversed at high feedrates. A follow–up paper will present experimental performance results from an implementation on a 3–axis CNC milling machine governed by a customized open–architecture software controller.

The focus of the present study has been on the IDC problem for machines with independently–controlled linear axes. It is of interest to adapt the methodology to more general problems, such as the cross–coupled control [3, 4, 15, 23, 25] of machine axes, in which *contour error* (geometrical path deviation) is penalized more severely than *feed error* (lag/lead along the path); control of 5–axis CNC machines, incorporating two rotary and three linear axes; and control of robotic devices involving multiple actuators and motion freedoms.

References

[1] Y. Altintas (2000), *Manufacturing Automation: Metal Cutting Mechanics, Machine Tool Vibrations, and CNC Design*, Cambridge University Press.

[2] J.–J. Chou and D. C. H. Yang (1991), Command generation for three–axis CNC machining, *ASME J. Eng. Indus.* **113**, 305–310.

[3] H–Y. Chuang and C–H. Liu (1991), Cross–coupled adaptive feedrate control for multi–axis machine tools, *ASME J. Dyn. Syst. Meas. Control* **113**, 451–457.

[4] J. R. Conway, C. A. Ernesto, R. T. Farouki, and M. Zhang (2012), Performance analysis of cross–coupled controllers for CNC machines based upon precise real–time contour error measurement, *Inter. J. Mach. Tools Manuf.* **52**, 30–39.

[5] K. Erkorkmaz, C–H. Yeung, and Y. Altintas (2006), Virtual CNC system. Part II. High speed contouring application, *Inter. J. Mach. Tools Manuf.* **46**, 1124–1138.

[6] C. A. Ernesto and R. T. Farouki (2010), Solution of inverse dynamics problems for contour error minimization in CNC machines, *Int. J. Adv. Manuf. Technol.* **49**, 589–604.

[7] R. T. Farouki (2008), *Pythagorean–Hodograph Curves: Algebra and Geometry Inseperable*, Springer, Berlin.

[8] R. T. Farouki, J. Manjunathaiah, D. Nicholas, G–F. Yuan, and S. Jee (1998), Variable–feedrate CNC interpolators for constant material removal rates along Pythagorean–hodograph curves, *Comput. Aided Design* **30**, 631–640.

[9] R. T. Farouki and T. Sakkalis (1991), Real rational curves are not "unit speed," *Comput. Aided Geom. Design* **8**, 151–157.

[10] R. T. Farouki and T. Sakkalis (2007), Rational space curves are not "unit speed," *Comput. Aided Geom. Design* **24**, 238–240.

[11] R. T. Farouki and S. Shah (1996), Real–time CNC interpolators for Pythagorean–hodograph curves, *Comput. Aided Geom. Design* **13**, 583–600.

[12] R. T. Farouki and Y–F. Tsai (2001), Exact Taylor series coefficients for variable–feedrate CNC curve interpolators, *Comput. Aided Design* **33**, 155–165.

[13] J.–T. Huang and D. C. H. Yang (1992), A generalized interpolator for command generation of parametric curves in computer–controlled machines, Proc. *Japan/USA Symposium on Flexible Automation*, Vol. 1, ASME, 393–399.

[14] R. Komanduri, K. Subramanian, and B. F. von Turkovich (eds.) (1984), *High Speed Machining*, PED–Vol. 12, ASME, New York.

[15] Y. Koren (1980), Cross–coupled biaxial computer control for manufacturing systems, *ASME J. Dyn. Syst. Meas. Control* **102**, 265–272.

[16] Y. Koren and C. C. Lo (1991), Variable–gain cross–coupling for contouring, *CIRP Ann.* **40**, 371–374.

[17] Y. Koren and C. C. Lo (1992), Advanced controllers for feed drives, *CIRP Ann.* **41**, 689–698.

[18] P. K. Kythe and M. R. Schäferkotter (2005), *Handbook of Computational Methods for Integration*, Chapman & Hall/CRC, Boca Raton.

[19] R–S. Lin and Y. Koren (1996), Real–time interpolators for multi–axis CNC machine tools, *Manuf. Syst.* **25**, 145–149.

[20] T. F. Schraeder and R. T. Farouki (2014), Experimental performance analysis of an inverse dynamics CNC compensation scheme for high–speed execution of curved toolpaths, *Int. J. Adv. Manuf. Technol.* **73**, 195–208.

[21] M. Shpitalni, Y. Koren, and C. C. Lo (1994), Realtime curve interpolators, *Comput. Aided Design* **26**, 832–838.

[22] S. Smith and J. Tlusty (1997), Current trends in high–speed machining, *ASME J. Manuf. Sci. Eng.* **119**, 664–666.

[23] K. Srinivasan and P. K. Kulkarni (1990), Cross–coupled control of biaxial feed drive servomechanisms, *ASME J. Dyn. Syst. Meas. Control* **112**, 225–232.

[24] D. J. Struik (1961), *Lectures on Classical Differential Geometry*, Dover Publications (reprint), New York.

[25] K–H. Su and M–Y. Cheng (2008), Contouring accuracy improvement using cross–coupled control and position error compensator, *Inter. J. Mach. Tools Manuf.* **48**, 1444–1453.

[26] J. Tlusty (1993), High–speed machining, *CIRP Ann.* **42**, 733–738.

[27] Y–F. Tsai, R. T. Farouki, and B. Feldman (2001), Performance analysis of CNC interpolators for time–dependent feedrates along PH curves, *Comput. Aided Geom. Design* **18**, 245–265.

[28] D. C. H. Yang and T. Kong (1994), Parametric interpolator versus linear interpolator for precision CNC machining, *Comput. Aided Design* **26**, 225–234.